

(43) 公開日 平成16年11月25日(2004.11.25)

テーマコード (参考)
5J104

審査請求 未請求 請求項の数 16 O L (全 32 頁)

(71) 出願人	399035766 エヌ・ティ・ティ・コミュニケーションズ 株式会社 東京都千代田区内幸町一丁目1番6号
(74) 代理人	100083806 弁理士 三好 秀和
(74) 代理人	100095500 弁理士 伊藤 正和
(74) 代理人	100101247 弁理士 高橋 俊一
(74) 代理人	100098327 弁理士 高松 俊雄

[最終頁に続く](#)

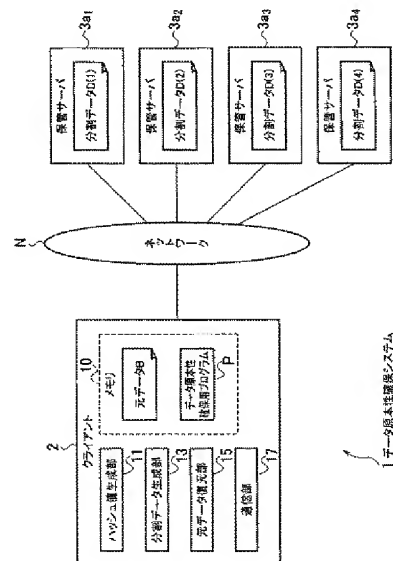
(54) 【発明の名称】 データ原本性確保方法およびシステム、ならびにデータ原本性確保用プログラム

(57) 【要約】

【課題】 データを、その原本性をより簡易に確保して保管することのできるデータ原本性確保方法およびシステム、ならびにデータ原本性確保用プログラムを提供する。

【解決手段】 元データBを、その原本性を確保して保管するデータ原本性確保システム1は、元データBを秘密分散法を用いて元データBが復元できる複数の分割データD(1)～D(4)にそれぞれ分割する分割データ生成部13と、分割された複数の分割データD(1)～D(4)をそれぞれ記憶するハードウェア的に独立した複数の保管サーバ3a1～3a4と、を備えている。

【選択図】 図1



【特許請求の範囲】**【請求項1】**

データを、その原本性を確保して保管するデータ原本性確保システムであって、

前記データを秘密分散法を用いて該データが復元できる複数の分割データにそれぞれ分割するデータ分割部と、

分割された複数の分割データをそれぞれ記憶するハードウェア的に独立した複数の保管部と、
を備えたことを特徴とするデータ原本性確保システム。

【請求項2】

前記データ分割部は、原本性確保対象となるデータ本体から、該データ本体をユニークに識別できる識別データを生成する識別データ生成部と、

生成された識別データと前記データ本体とを結合した前記データを秘密分散法を用いて分割する分割部と
を備えたことを特徴とする請求項1記載のデータ原本性確保システム。

【請求項3】

前記識別データ生成部は、前記識別データとして、前記データ本体を所定のハッシュ関数を用いてハッシュ値を生成する生成部を備えたことを特徴とする請求項2記載のデータ原本性確保システム。

【請求項4】

前記データ分割部は、分割された各分割データに、データ分割に関する管理情報を合わせたデータ全体から、該データ全体をユニークに識別できる分割識別データをそれぞれ生成し、

前記複数の保管部は、前記分割データ、前記データ分割に関する管理情報、および前記分割識別データをそれぞれ記憶することを特徴とする請求項2又は3記載のデータ原本性確保システム。

【請求項5】

前記複数の保管部から前記複数の分割データをそれぞれ読み出し、読み出した複数の分割データから秘密分散法により前記データ本体および前記識別データをそれぞれ復元するデータ復元部と、

復元されたデータ本体から前記識別データを再生成し、再生成された識別データと前記復元された識別データとが一致するか否かを確認する確認部と、
を備えたことを特徴とする請求項2又は3記載のデータ原本性確保システム。

【請求項6】

前記複数の保管部から前記分割データ、前記データ分割に関する管理情報、および前記分割識別データをそれぞれ読み出し、読み出した複数の分割データから秘密分散法により前記データ本体および前記識別データをそれぞれ復元するデータ復元部と、

復元されたデータ本体から前記識別データを再生成し、再生成された識別データと前記復元された識別データとが一致するか否かを確認する第1の確認部と、

前記複数の分割データそれぞれに対して、読み出した分割データおよびデータ分割に関する管理情報とを合わせたデータ全体から、前記分割識別データを再生成し、再生成された分割識別データと、読み出した前記分割識別データとが一致するか否かを確認する第2の確認部と、
を備えたことを特徴とする請求項4記載のデータ原本性確保システム。

【請求項7】

前記複数の保管部から前記複数の分割データをそれぞれ読み出し、読み出した複数の分割データを、異なる組合せにより組み合わせて複数のデータを復元し、復元した複数のデータがそれぞれ一致するか否かを確認する復元確認部をさらに備えたことを特徴とする請求項1記載のデータ原本性確保システム。

【請求項8】

データを、その原本性を確保して保管するデータ原本性確保方法であって、

前記データを秘密分散法を用いて該データが復元できる複数の分割データにそれぞれ分割するステップと、

分割された複数の分割データをハードウェア的に独立した複数の保管装置にそれぞれ記憶するステップと、
を備えたことを特徴とするデータ原本性確保方法。

【請求項9】

前記分割ステップは、原本性確保対象となるデータ本体から、該データ本体をユニークに識別できる識別データを生成するステップと、

生成された識別データと前記データ本体とを結合した前記データを秘密分散法を用いて分割するステップと、
を備えたことを特徴とする請求項8記載のデータ原本性確保方法。

【請求項10】

データを、その原本性を確保して保管するためのコンピュータが読み取り可能なデータ原本性確保用プログラムであって、

前記コンピュータに、前記データを秘密分散法を用いて該データが復元できる複数の分割データにそれぞれ分割するデータ分割処理と、

分割された複数の分割データを、該コンピュータが通信可能なハードウェア的に独立した複数の保管装置にそれぞれ記憶する保管処理と、
をそれぞれ実行させることを特徴とするデータ原本性確保用プログラム。

【請求項11】

前記分割処理は、原本性確保対象となるデータ本体から、該データ本体をユニークに識別できる識別データを生成する識別データ生成処理と、

生成された識別データと前記データ本体とを結合した前記データを秘密分散法を用いて分割する分割処理と、
を備えたことを特徴とする請求項10記載のデータ原本性確保用プログラム。

【請求項12】

前記識別データ生成処理は、前記識別データとして、前記データ本体を所定のハッシュ関数を用いてハッシュ値を生成することを特徴とする請求項11記載のデータ原本性確保用プログラム。

【請求項13】

前記データ分割処理は、分割された各分割データに、データ分割に関する管理情報を合わせたデータ全体から、該データ全体をユニークに識別できる分割識別データをそれぞれ生成し、

前記保管処理は、前記複数の保管装置に、前記分割データ、前記データ分割に関する管理情報、および前記分割識別データをそれぞれ記憶することを特徴とする請求項11又は12記載のデータ原本性確保用プログラム。

【請求項14】

前記複数の保管装置から前記複数の分割データをそれぞれ読み出し、読み出した複数の分割データから秘密分散法により前記データ本体および前記識別データをそれぞれ復元するデータ復元処理と、

復元されたデータ本体から前記識別データを再生成し、再生成された識別データと前記復元された識別データとが一致するか否かを確認する確認処理と、
を前記コンピュータに実行させることを特徴とする請求項11又は12記載のデータ原本性確保用プログラム。

【請求項15】

前記複数の保管装置から前記分割データ、前記データ分割に関する管理情報、および前記分割識別データをそれぞれ読み出し、読み出した複数の分割データから秘密分散法により前記データ本体および前記識別データをそれぞれ復元するデータ復元処理と、

復元されたデータ本体から前記識別データを再生成し、再生成された識別データと前記復元された識別データとが一致するか否かを確認する第1の確認処理と、

前記複数の分割データそれぞれに対して、読み出した分割データおよびデータ分割に関する管理情報とを合わせたデータ全体から、前記分割識別データを再生成し、再生成された分割識別データと、読み出した前記分割識別データとが一致するか否かを確認する第2の確認処理と、
を前記コンピュータに実行させることを特徴とする請求項13記載のデータ原本性確保用プログラム。

【請求項16】

前記複数の保管装置から前記複数の分割データをそれぞれ読み出し、読み出した複数の分割データを、異なる組合せにより組み合わせて複数のデータを復元し、復元した複数のデータがそれぞれ一致するか否かを確認する復元確認処理をさらに前記コンピュータに実行させることを特徴とする請求項10記載のデータ原本性確保用プログラム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、データを、その原本性を確保して保管するためのデータ原本性確保方法およびシステム、ならびにデータ原本性確保用プログラムに関する。

【背景技術】

【0002】

I Tの発展に伴って、自らの端末をインターネット等の通信ネットワークに接続し、この通信ネットワークに接続されている他端末に対して通信ネットワークを介してデータを通信する機会が非常に増えている。

【0003】

このとき、端末は通信ネットワークに接続されているため、悪意を持った第三者（ハッカー等）が通信ネットワークに接続された第三者側の端末を介して端末にアクセスし、端末に保管されているデータを改竄する恐れが生じており、改竄されたデータの原本性（故意または過失による改竄がなく、内容が完全であること）を確保する必要が生じていた。

【0004】

この点、従来では、電子署名を用いてデータの改竄を検出可能にして、原本性を確保する技術が導入されている（例えば、非特許文献1参照）。

【0005】

すなわち、電子署名技術を用いた原本性確保方法によれば、データ送信元のA氏は、予め認証局（CA; Certification Authority）により保証された公開鍵入りの電子証明書を受取る。

【0006】

そして、A氏は、送信したいデータ（データ本体とも呼ぶ）について電子署名を作成し、作成した電子署名を含むデータ（データ本体+電子署名+電子証明書）をB氏に送信する。

【0007】

B氏は、受信したデータのうち、データ本体から生成したハッシュ値と、A氏から受け取ったA氏の電子証明書に含まれる公開鍵で電子署名を復号化して得た値とを比較することにより、データが改竄されたか否かを判断している。

【非特許文献1】”PKI 関連技術説明”、[online]、平成14年12月12日、情報処理振興事業協会、[平成15年4月11日検索]、インターネット<URL: <http://www.ipa.go.jp/security/pki/index.html>>

【発明の開示】

【発明が解決しようとする課題】

【0008】

しかしながら、電子署名技術を用いた原本性確保方法によれば、B氏側において受け取ったデータを長期間保管する場合、秘密鍵の危殆化等からデータが改竄される恐れが生じている。

【0009】

そこで、電子署名技術を用いた原本性確保方法において、認証局から発行される電子証明書には予め有効期限が定められており、有効期限到来前にA氏側において改めて秘密鍵と公開鍵とを生成し、公開鍵について電子証明書を認証局から発行してもらい、新しい秘密鍵を用いて再度電子署名付きのデータを作成する必要がある。

【0010】

しかしながら、上述した電子証明書の有効期限に基づく認証局からの電子証明書の再発行処理、および新しい秘密鍵を用いた電子署名付きデータ作成処理は非常に手間および時間がかかり、また、有効期限毎に上述した電子証明書の再発行処理および電子署名付きデータ作成処理を繰り返す行わなければならない、データを、その原本性をより簡易に確保しながら保管できる方法の開発が待望されていた。

【0011】

本発明は、上述した事情に鑑みてなされたものであり、データを、その原本性をより簡易に確保して保管することのできるデータ原本性確保方法およびシステム、ならびにデータ原本性確保プログラムを提供することをその目的とする。

【課題を解決するための手段】

【0012】

上述した目的を達成するため、本発明は、請求項1に記載したように、データを、その原本性を確保して保管するデータ原本性確保システムであって、前記データを秘密分散法を用いて該データが復元できる複数の分割データにそれぞれ分割するデータ分割部と、分割された複数の分割データをそれぞれ記憶するハードウェア的に独立した複数の保管部と、を備えたことを要旨とする。

【0013】

請求項2に記載した発明は、前記データ分割部は、原本性確保対象となるデータ本体から、該データ本体をユニークに識別できる識別データを生成する識別データ生成部と、生成された識別データと前記データ本体とを結合した前記データを秘密分散法を用いて分割する分割部とを備えたことを要旨とする。

【0014】

請求項3に記載した発明は、前記識別データ生成部は、前記識別データとして、前記データ本体を所定のハッシュ関数を用いてハッシュ値を生成する生成部を備えたことを要旨とする。

【0015】

請求項4に記載した発明は、前記データ分割部は、分割された各分割データに、データ分割に関する管理情報を合わせたデータ全体から、該データ全体をユニークに識別できる分割識別データをそれぞれ生成し、前記複数の保管部は、前記分割データ、前記データ分割に関する管理情報、および前記分割識別データをそれぞれ記憶することを要旨とする。

【0016】

請求項5に記載した発明は、前記複数の保管部から前記複数の分割データをそれぞれ読み出し、読み出した複数の分割データから秘密分散法により前記データ本体および前記識別データをそれぞれ復元するデータ復元部と、復元されたデータ本体から前記識別データを再生成し、再生成された識別データと前記復元された識別データとが一致するか否かを確認する確認部と、を備えたことを要旨とする。

【0017】

請求項6に記載した発明は、前記複数の保管部から前記分割データ、前記データ分割に関する管理情報、および前記分割識別データをそれぞれ読み出し、読み出した複数の分割データから秘密分散法により前記データ本体および前記識別データをそれぞれ復元するデータ復元部と、復元されたデータ本体から前記識別データを再生成し、再生成された識別データと前記復元された識別データとが一致するか否かを確認する第1の確認部と、前記複数の分割データそれぞれに対して、読み出した分割データおよびデータ分割に関する管理情報とを合わせたデータ全体から、前記分割識別データを再生成し、再生成された分割

識別データと、読み出した前記分割識別データとが一致するか否かを確認する第2の確認部と、を備えたことを要旨とする。

【0018】

請求項7に記載した発明は、前記複数の保管部から前記複数の分割データをそれぞれ読み出し、読み出した複数の分割データを、異なる組合せにより組み合わせて複数のデータを復元し、復元した複数のデータがそれぞれ一致するか否かを確認する復元確認部をさらに備えたことを要旨とする。

【0019】

請求項8に記載した発明は、データを、その原本性を確保して保管するデータ原本性確保方法であって、前記データを秘密分散法を用いて該データが復元できる複数の分割データにそれぞれ分割するステップと、分割された複数の分割データをハードウェア的に独立した複数の保管装置にそれぞれ記憶するステップと、を備えたことを要旨とする。

【0020】

請求項9に記載した発明は、前記分割ステップは、原本性確保対象となるデータ本体から、該データ本体をユニークに識別できる識別データを生成するステップと、生成された識別データと前記データ本体とを結合した前記データを秘密分散法を用いて分割するステップと、を備えたことを要旨とする。

【0021】

請求項10に記載した発明は、データを、その原本性を確保して保管するためのコンピュータが読み取り可能なデータ原本性確保プログラムであって、前記コンピュータに、前記データを秘密分散法を用いて該データが復元できる複数の分割データにそれぞれ分割するデータ分割処理と、分割された複数の分割データを、該コンピュータが通信可能なハードウェア的に独立した複数の保管装置にそれぞれ記憶する保管処理と、をそれぞれ実行させることを要旨とする。

【0022】

請求項11に記載した発明は、前記分割処理は、原本性確保対象となるデータ本体から、該データ本体をユニークに識別できる識別データを生成する識別データ生成処理と、生成された識別データと前記データ本体とを結合した前記データを秘密分散法を用いて分割する分割処理と、を備えたことを要旨とする。

【0023】

請求項12に記載した発明は、前記識別データ生成処理は、前記識別データとして、前記データ本体を所定のハッシュ関数を用いてハッシュ値を生成することを要旨とする。

【0024】

請求項13に記載した発明は、前記データ分割処理は、分割された各分割データに、データ分割に関する管理情報を合わせたデータ全体から、該データ全体をユニークに識別できる分割識別データをそれぞれ生成し、前記保管処理は、前記複数の保管装置に、前記分割データ、前記データ分割に関する管理情報、および前記分割識別データをそれぞれ記憶することを要旨とする。

【0025】

請求項14に記載した発明は、前記複数の保管装置から前記複数の分割データをそれぞれ読み出し、読み出した複数の分割データから秘密分散法により前記データ本体および前記識別データをそれぞれ復元するデータ復元処理と、復元されたデータ本体から前記識別データを再生成し、再生成された識別データと前記復元された識別データとが一致するか否かを確認する確認処理と、を前記コンピュータに実行させることを要旨とする。

【0026】

請求項15に記載した発明は、前記複数の保管装置から前記分割データ、前記データ分割に関する管理情報、および前記分割識別データをそれぞれ読み出し、読み出した複数の分割データから秘密分散法により前記データ本体および前記識別データをそれぞれ復元するデータ復元処理と、復元されたデータ本体から前記識別データを再生成し、再生成された識別データと前記復元された識別データとが一致するか否かを確認する第1の確認処理

と、前記複数の分割データそれぞれに対して、読み出した分割データおよびデータ分割に関する管理情報とを合わせたデータ全体から、前記分割識別データを再生成し、再生成された分割識別データと、読み出した前記分割識別データとが一致するか否かを確認する第2の確認処理と、を前記コンピュータに実行させることを要旨とする。

【0027】

請求項16に記載した発明は、前記複数の保管装置から前記複数の分割データをそれぞれ読み出し、読み出した複数の分割データを、異なる組合せにより組み合わせて複数のデータを復元し、復元した複数のデータがそれぞれ一致するか否かを確認する復元確認処理をさらに前記コンピュータに実行させることを要旨とする。

【発明の効果】

【0028】

本発明に係わるデータ原本性確保方法およびシステム、ならびにデータ原本性確保用プログラムによれば、データを、秘密分散法を用いて複数の分割データに分割してハードウェア的に独立した複数の保管部（保管装置）にそれぞれ保管しているため、そのデータの内の少なくとも一部が改竄された場合でも、改竄データを含む分割データから、その分割データに対する改竄の有無を容易に確認することができる。

【0029】

この結果、例えばデータを長期間に亘って保管する場合であっても、電子署名のように、電子証明書の再発行処理および電子署名付きデータ作成処理を繰り返すことなく、簡易にデータを保管することができる。

【発明を実施するための最良の形態】

【0030】

本発明に係わるデータ原本性確保方法およびシステム、ならびにデータ原本性確保用プログラムの実施の形態について、添付図面を参照して説明する。

【0031】

（第1の実施の形態）

図1は、本発明の第1の実施の形態に係わるデータ原本性確保システム1の概略構成を示すブロック図である。

【0032】

図1に示すように、データ原本性確保システム1は、インターネット等の通信用ネットワークNに対して接続して通信できるクライアントコンピュータ（以下、単にクライアントとする）2と、ネットワークNに対して通信可能に接続されており、互いにハードウェア的に独立した複数（本実施の形態では4とする）のデータ保管用サーバコンピュータ（以下、単に保管サーバとする）3a1～3a4とを備えている。

【0033】

クライアント2は、演算処理部であるCPU、このCPUに接続され該CPUおよびネットワークN間の通信を可能にするインタフェース、CPUに接続されたデータ入力用の入力部およびCPUに接続されたメモリ10をそれぞれ備えている。

【0034】

メモリ10には、原本性を確保して保管する対象となるデータ（例えば、M（Mは自然数）バイトのデジタルデータ；以下、元データとも呼ぶ）Bが格納されている。また、メモリ10には、クライアント2（そのCPU）が読み取り可能であり、後述する図2および図3に示す原本性確保処理（データ保管処理）をクライアント2（そのCPU）に実行させるためのデータ原本性確保用プログラムPが搭載されている。

【0035】

なお、このデータ原本性確保用プログラムPは、磁気メモリや半導体メモリ等の各種記録媒体に搭載され、必要に応じてその記録媒体からクライアント2に読み出されてメモリ10にロードされるように構成することも可能である。

【0036】

クライアント2は、データ原本性確保用プログラムPにより実現される機能として、メ

メモリ 10 に格納された元データ B から、その元データ B をユニークに識別できる例えばハッシュ値を生成するハッシュ値生成部 11 と、元データ B およびハッシュ値 H を含むデータを秘密分散法を用いて分割して分割データ D (1) ～ D (4) (本実施形態では、分割数を 4 とする) を生成する分割データ生成部 13 と、分割データ D (1) ～ D (4) から元データ B およびハッシュ値 H を生成する元データ復元部 15 と、分割データ D (1) ～ D (4) をネットワーク N を介して通信する通信部 17 とを備えている。

【0037】

各保管サーバ 3 a 1 ～ 3 a 4 は、演算処理部である CPU、この CPU に接続され該 CPU およびネットワーク N 間の通信を可能にするインタフェース、CPU に接続されたデータ入力用の入力部、CPU に接続されたメモリおよびハードディスク等の記憶装置をそれぞれ備えている。

【0038】

次に、本実施の形態に係わるデータ原本性確保システム 1 の全体処理について説明する。

【0039】

図 2 および図 3 に示すように、クライアント 2 は、データ原本性確保用プログラム P に従って動作し、ハッシュ値生成部 11 の機能として、メモリ 10 に記憶された、原本性を確保して保管したい元データ B をメモリ 10 から読み込み、読み込んだ元データ B を、所定のハッシュ関数を用いてその元データ B のハッシュ値 H を生成する (ステップ S 1) 。

【0040】

このハッシュ値 H は、その元データ B が 1 ビットでも変更されると全く異なる値を示す性質、すなわち、元データ B をユニークに識別できる性質を有している。

【0041】

次いで、クライアント 2 は、分割データ生成部 13 の機能として、生成したハッシュ値 H を含む元データ B を秘密分散法を用いて 4 つのデータ (分割データ) D (1) ～ D (4) に分割する (ステップ S 2) 。

【0042】

以下、ステップ S 2 の秘密分散法に基づく分割データ D (1) ～ D (4) 生成処理について詳細に説明する。

【0043】

例えば、2 次多項式 $F(x) = ax^2 + bx + B \pmod{p}$; p で割った時の余りを表す) を基にした Shamir の秘密分散法 { (k , n) 閾値法 ; 但し、分割数を表す n を 4 とし、復元できる数を表す k を 3 とする } で考える。ここで B は元データ、 $F(x)$ は分割データである。 a 、 b 、 p は、元データ B の分割に際して任意に決定される。但し、 p は、 a 、 b 、 B よりも大きい素数とする。

【0044】

このとき、クライアント 2 の分割データ生成処理により、分割データ $F(1) \sim F(4)$ { 上記分割データ $D(1) \sim D(4)$ に対応 } は、次式 (1) ～ (4) のように作成される。

【0045】

$$F(1) = a + b + B \pmod{p} \quad \cdots (1)$$

$$F(2) = 4a + 2b + B \pmod{p} \quad \cdots (2)$$

$$F(3) = 9a + 3b + B \pmod{p} \quad \cdots (3)$$

$$F(4) = 16a + 4b + B \pmod{p} \quad \cdots (4)$$

この分割データ $F(1) \sim F(4)$ の内、 $k=3$ 以上の分割データ { 例えば、 $F(1)$ 、 $F(2)$ 、 $F(4)$ } が集まれば、この分割データ

$$F(1) = a + b + B \pmod{p} \quad \cdots (1)$$

$$F(2) = 4a + 2b + B \pmod{p} \quad \cdots (2)$$

$$F(4) = 16a + 4b + B \pmod{p} \quad \cdots (4)$$

を連立して元データ B を求めることができる。

【0046】

そして、 $k-1$ 以下の分割データが集まっても、元データBを復元することはできない。

【0047】

なお、元データBが長いデータ列である場合には、クライアント2は、例えば元データBの先頭から1バイト毎に $F(1)$ から $F(4)$ を順次作成して分割データ $D(1)$ ($F(1)$) \sim $D(4)$ ($F(4)$)を作成する。

【0048】

そして、クライアント2は、通信部17の機能として、作成した分割データ $D(1) \sim D(4)$ を保管サーバ3a1 \sim 3a4にネットワークNを介してそれぞれ送信する(ステップS3)。

【0049】

各保管サーバ3a1 \sim 3a4は、ネットワークNを介して送信されてきた分割データ $D(1) \sim D(4)$ を、それぞれのハードディスク等の記憶装置に記憶する(ステップS5)。

【0050】

このようにして、元データBを、そのハッシュ値Hを含んで分割された分割データ $D(1) \sim D(4)$ として保管サーバ3a1 \sim 3a4に保管することができる。

【0051】

次に、保管サーバ3a1 \sim 3a4に保管された分割データ $D(1) \sim D(4)$ が改竄されているか否かを確認する場合、クライアント2は、分割データ $D(1) \sim D(4)$ のダウンロード要求を保管サーバ3a1 \sim 3a4にそれぞれ送信する(ステップS10)。

【0052】

各保管サーバ3a1 \sim 3a4は、送信されてきたダウンロード要求に応じて、それぞれのハードディスク等の記憶装置に保管された各分割データ $D(1) \sim D(4)$ を各記憶装置から読み出し、読み出した各分割データ $D(1) \sim D(4)$ をネットワークNを介してクライアント2に送信する(ステップS11)。

【0053】

クライアント2は、データ原本性確保用プログラムPに従って動作し、元データ復元部15の機能として、ネットワークNを介して送信されてきた分割データ $D(1) \sim D(4)$ を受信し、受信した分割データ $D(1) \sim D(4)$ に基づいて秘密分散法により元データB1およびハッシュ値H1をそれぞれ復元する(ステップS12)。

【0054】

次いで、クライアント2は、元データ復元部15の機能として、復元した元データB1からハッシュ値H2を再生成し、再生成したハッシュ値H2と復元したハッシュ値H1とを比較して一致するか否かを確認する(ステップS13)。

【0055】

このステップS12およびS13の処理を具体的に説明する。

【0056】

例えば、上記分割データ $F(1)$ 、 $F(2)$ 、 $F(3)$ 、 $F(4)$ の内、一部の分割データが改竄(例えば、 $F(2) \sim F(4)$ が $Fa(2) \sim Fa(4)$ に改竄)されたと仮定する。

【0057】

このとき、クライアント2は、下式

$$F(1) = a + b + B \pmod{p} \quad \dots (1)$$

$$Fa(2) = 4a + 2b + B \pmod{p} \quad \dots (2)$$

$$Fa(3) = 9a + 3b + B \pmod{p} \quad \dots (3)$$

$$Fa(4) = 16a + 4b + B \pmod{p} \quad \dots (4)$$

の中から少なくとも3つの式を連立させてハッシュ値Hを含む元データBを復元することになる。

【0058】

しかしながら、分割データF(2)～F(4)が分割データFa(2)～Fa(4)に改竄されているため、この分割データFa(2)～Fa(4)を連立しても、復元された元データB1は、原本性確保対象となる元データBとは異なるため、その元データB1から再生成されたハッシュ値H2も、復元されたハッシュ値H1とは異なる。この結果、保管サーバ3a1～3a4に保管されている分割データD(1)～D(4)の少なくとも一部が改竄されていることをクライアント2側において確認することができる。

【0059】

以上述べたように、本実施形態によれば、元データBを、秘密分散法を用いて複数の分割データD(1)～D(4)に分割して保管サーバ3a1～3a4にそれぞれ保管しているため、その分割データD(1)～D(4)の内の少なくとも一部が改竄された場合でも、改竄データを含む分割データD(1)～D(4)から、その分割データD(1)～D(4)に対する改竄の有無を容易に確認することができる。

【0060】

この結果、例えば元データBを長期間に亘って保管する場合であっても、電子署名のように、電子証明書の再発行処理および電子署名付きデータ作成処理を繰り返すことなく、簡易に元データBを保管することができる。

【0061】

また、本実施形態によれば、元データBを分割データD(1)～D(4)に分割して保管サーバ3a1～3a4に保管しているため、例えば、分割データD(1)～D(4)の内の1つの分割データが改竄された場合でも、残りの分割データを用いて元データBを復元することができる。

【0062】

この結果、元データBの原本性をより確実に確保することができる。

【0063】

(第2の実施の形態)

図4は、本発明の第2の実施の形態に係わるデータ原本性確保システムにおけるデータ原本性確保プログラムに基づく原本性確保処理の一例を示す概略フローチャートである。なお、本実施の形態におけるデータ原本性確保システムの構成については、そのデータ原本性確保プログラムの内容およびクライアントの処理が異なり、他の構成については、図1に示す構成と略同等であるため、その説明は省略する。

【0064】

本実施の形態において、クライアント2は、分割データ生成部13の機能として、メモリ10に記憶された、原本性を確保して保管したい元データBをメモリ10から読み込み、読み込んだ元データBを秘密分散法を用いて4つのデータ(分割データ)D(1)～D(4)に分割する(ステップS21)。

【0065】

なお、ステップS21の処理は、ステップS2の処理と略同等である。

【0066】

次いで、クライアント2は、通信部17の機能として、作成した分割データD(1)～D(4)を保管サーバ3a1～3a4にネットワークNを介してそれぞれ送信する(ステップS22)。

【0067】

各保管サーバ3a1～3a4は、ネットワークNを介して送信されてきた分割データD(1)～D(4)を、それぞれのハードディスク等の記憶装置に記憶する(ステップS23)。

【0068】

このようにして、元データBを、分割データD(1)～D(4)として保管サーバ3a1～3a4に保管することができる。

【0069】

次に、保管サーバ3 a 1～3 a 4に保管された分割データD (1)～D (4)が改竄されているか否かを確認する場合、クライアント2は、分割データD (1)～D (4)のダウンロード要求を保管サーバ3 a 1～3 a 4にそれぞれ送信する(ステップS 3 0)。

【0070】

各保管サーバ3 a 1～3 a 4は、送信されてきたダウンロード要求に応じて、それぞれのハードディスク等の記憶装置に保管された各分割データD (1)～D (4)を、その記憶装置から読み出し、読み出した各分割データD (1)～D (4)をネットワークNを介してクライアント2に送信する(ステップS 3 1)。

【0071】

クライアント2は、元データ復元部1 5の機能として、ネットワークNを介して送信されてきた分割データD (1)～D (4)を受信し、受信した分割データD (1)～D (4)を秘密分散法により異なる組合せにより組み合わせて複数の元データB 1～B m(本実施の形態では、m=2)を復元し、復元した元データB 1～B 2が互いに一致するか否かを確認する(ステップS 3 2)。

【0072】

このステップS 3 2の処理を具体的に説明する。

【0073】

クライアント2は、上記分割データF (1)、F (2)、F (3)、F (4)の内、例えば、所定の組合せ、例えば分割データF (1)～F (3)を用いて秘密分散法により元データB 1を復元し、次いで、上記組合せとは異なる組合せ、例えば、分割データF (2)～F (4)を用いて秘密分散法により元データB 2を復元する。

【0074】

このとき、上記分割データF (1)～F (4)の内の一部に改竄が発生した場合、上述したように、復元された元データB 1およびB 2は、原本性確保対象となる元データBとは異なり、かつ元データB 1およびB 2は分割データF (1)～F (4)を互いに異なる組合せにより組み合わせているため、復元した元データB 1およびB 2は互いに異なる。

【0075】

したがって、クライアント2は、復元した元データB 1および元データB 2との不一致により、保管サーバ3 a 1～3 a 4に保管されている分割データD (1)～D (4)の少なくとも一部が改竄されていることをクライアント2側において確認することができる。

【0076】

なお、本実施の形態では、分割データF (1)～F (4)の中から異なる組合せとして、分割データF (1)～F (3)と分割データF (2)～F (4)とをそれぞれ選択したが、本発明はこの組合せに限定されるものではなく、4つの分割データD (1)～D (4)それぞれが少なくとも1回復元に使用される複数の組合せを選択すればよく、互いの組合せに基づいて復元された元データが一致しているか否かを確認することができる。

【0077】

(第3の実施の形態)

図5は、本発明の第3の実施の形態に係わるデータ原本性確保システム4の概略構成を示すブロック図である。

【0078】

図5に示すように、データ原本性確保システム4は、インターネット等の通信用ネットワークNに対して接続して通信できる端末5および分割装置6と、ネットワークNに対して通信可能に接続されており、互いにハードウェア的に独立した複数(本実施の形態では3とする)のデータ保管用サーバコンピュータ(以下、単に保管サーバとする)3 a 1～3 a 3とを備えている。そして、上記システム構成により、分割装置6は、ネットワークNを介して端末5からのデータ保管要求に応じてデータを複数の分割データに分割し、この分割した複数の分割データをネットワークNを介して複数の保管サーバ3 a 1～3 a 3に保管するようになっている。尚、端末5と分割装置6との間の通信、および分割装置6と各保管サーバ3 a 1～3 a 3との間の通信は、通信内容の漏洩を防止するため、SSL

(Secure Socket Layer)、I P-V P N (Virtual Private Network) などにより、通信データが暗号化されている。

【0079】

端末5および分割装置6は、演算処理部であるCPU、このCPUに接続され該CPUおよびネットワークN間の通信を可能にするインタフェース、CPUに接続されたデータ入力用の入力部およびCPUに接続されたメモリをそれぞれ備えている。

【0080】

端末5は、利用者が原本性を確保して保管するデータ（例えば、M（Mは自然数）バイトのデジタルデータ；以下、元データとも呼ぶ）Bを分割装置6に送信するとともに、保管したデータを分割装置6から受信するデータ送受信部31を備えている。

【0081】

分割装置6のメモリ10には、端末5から送信された元データBが格納される。また、メモリ10には、分割装置6（そのCPU）が読み取り可能であり、後述する図6および図7に示すデータ原本性確保処理を分割装置6（そのCPU）に実行させるためのデータ原本性確保プログラムP3が搭載されている。

【0082】

なお、このデータ原本性確保プログラムP3は、磁気メモリや半導体メモリ等の各種記録媒体に搭載され、必要に応じてその記録媒体から分割装置6に読み出されてメモリにロードされるように構成することも可能である。

【0083】

分割装置6は、データ原本性確保プログラムP3により実現される機能として、メモリ10に格納された元データBから、その元データBをユニークに識別できる例えばハッシュ値Hを生成するとともに、後述するハッシュ値h1～h3を生成するハッシュ値生成部32と、元データBおよびハッシュ値Hを含むデータを後述する秘密分散法（第1および第2の実施の形態とは異なる秘密分散法）Sを用いて分割して分割データ（以下、分割データ本体ともよぶ）D（1）～D（3）（本実施形態では、分割数を3とする）を生成するとともに、該分割データ本体D（1）～D（3）、ヘッダ情報、および分割データ本体とヘッダ情報の内容全体から生成されたハッシュ値h1～h3から構成された分割データ（以下、送信分割データともよぶ）DD（1）～DD（3）を生成する分割データ生成部33と、分割データ本体D（1）～D（3）を生成する際に用いられる乱数データを発生させる乱数発生部34と、保管サーバ3a1～3a3から取得した分割データ本体D（1）～D（3）およびそのヘッダ情報からハッシュ値h1～h3を再生成するとともに、復元された元データBからハッシュ値Hを再生成し、データの真正性を確認するハッシュ値確認部35と、分割データ本体D（1）～D（3）から元データBおよびハッシュ値Hを秘密分散法Sを用いて生成する元データ復元部36と、元データBおよび送信分割データDD（1）～DD（3）をネットワークNを介して送受信するデータ送受信部37とを備えている。

【0084】

各保管サーバ3a1～3a3は、演算処理部であるCPU、このCPUに接続され該CPUおよびネットワークN間の通信を可能にするインタフェース、CPUに接続されたデータ入力用の入力部およびCPUに接続されたメモリおよびハードディスク等の記憶装置をそれぞれ備えている。

【0085】

次に、本実施の形態に係わるデータ原本性確保システム4の全体処理を図6および図7を用いて説明する。ここで、図6は、本実施の形態におけるデータ原本性確保プログラムP3に基づくデータ原本性確保処理を示すフローチャートであり、図7は、図6に示すデータ原本性確保処理のデータシーケンスである。

【0086】

図6および図7に示すように、利用者が保管したい元データBを指定し、端末5から分割装置6に該元データBを送信すると、分割装置6は、データ原本性確保プログラムP

3に従って動作し、ハッシュ値生成部32の機能として、メモリ10に記憶された、原本性を確保して保管したい元データBをメモリ10から読み込み、読み込んだ元データBを、所定のハッシュ関数を用いてその元データBのハッシュ値Hを生成する（ステップS101, S102）。

【0087】

このハッシュ値Hは、その元データBが1ビットでも変更されると全く異なる値を示す性質、すなわち、元データBをユニークに識別できる性質を有している。

【0088】

次いで、分割装置6は、分割データ生成部33の機能として、生成したハッシュ値Hを含む元データBを秘密分散法Sを用いて3つのデータ（分割データ）D（1）～D（3）に分割する（ステップS103）。そして、各分割データD（i）（i=1, 2, 3）について、図8に示す形式のデータ、即ち、ハッシュ値h_i（i=1, 2, 3）、データ分割に関する管理情報を含むヘッダ情報A_i（i=1, 2, 3）、および分割データD（i）（i=1, 2, 3）から構成される送信分割データDD（i）を生成する（ステップS104, S105）。

【0089】

ここで、ハッシュ値h_iは、ハッシュ値生成部32の機能として、ヘッダ情報A_iおよび分割データD（i）を合わせた内容全体から、所定のハッシュ関数を用いて生成されるものである。尚、ヘッダ情報A_iのデータ識別子A_{i1}には、本実施の形態の秘密分散法Sにより分割されたデータであることを示す情報、バージョン情報A_{i2}には、データフォーマットのバージョンを識別する情報、およびヘッダサイズA_{i3}には、元データサイズから分割データサイズまでの領域の合計データの長さを示す情報が格納されるようになっている。また、元データサイズA_{i4}は、元データ（ハッシュ値Hを含む元データB）の長さを示す情報、分割数A_{i5}には、データの分割数（本実施の形態においては3）を示す情報、および処理単位ビット長A_{i6}には、秘密分散法Sによる分割処理において用いられる後述する処理単位ビット長を示す情報が格納されるようになっている。また、乱数データサイズA_{i7}には、秘密分散法Sによる分割処理において用いられる乱数データの長さを示す情報、分割データタイプA_{i8}には、いくつ目の分割データであるかを示す情報（例えば、3分割の場合は、分割データがD（1）、D（2）、D（3）のいずれかであるかを示す情報）、および分割データサイズA_{i9}には、分割データD（i）の長さを示す情報が格納されるようになっている。

【0090】

そして、分割装置6は、データ送受信部37の機能として、作成した送信分割データD（1）～DD（3）を保管サーバ3a1～3a3にネットワークNを介してそれぞれ送信する（ステップS106）。

【0091】

各保管サーバ3a1～3a3は、ネットワークNを介して送信されてきた送信分割データDD（1）～DD（3）を、それぞれのハードディスク等の記憶装置に記憶する（ステップS107）。

【0092】

このようにして、元データBをそのハッシュ値Hを含んで分割した分割データD（1）～D（3）、ヘッダ情報A1～A3、およびハッシュ値h1～h3から構成される送信分割データDD（1）～DD（3）をそれぞれ保管サーバ3a1～3a3に保管することができる。

【0093】

次に、保管サーバ3a1～3a3に保管された送信分割データDD（1）～DD（3）を取り出す場合には、利用者が取り出したい元データBを指定し、端末5から分割装置6に取り出し指示を送信すると、分割装置6は、送信分割データDD（1）～DD（3）のダウンロード要求を保管サーバ3a1～3a3にそれぞれ送信する（ステップS108, S109）。

【0094】

各保管サーバ3 a 1～3 a 3は、送信されてきたダウンロード要求に応じて、それぞれのハードディスク等の記憶装置に保管された各送信分割データDD (1)～DD (3)を各記憶装置から読み出し、読み出した各送信分割データDD (1)～DD (3)をネットワークNを介して分割装置6に送信する(ステップS 1 1 0)。

【0095】

分割装置6は、データ原本性確保用プログラムP 3に従って動作し、ハッシュ値確認部3 5の機能として、ネットワークNを介して送信されてきた送信分割データDD (1)～DD (3)を受信し、受信した送信分割データDD (1)～DD (3)の分割データ本体D (1)～D (3)およびヘッダ情報A 1～A 3からハッシュ値h 1'～h 3'を再生成する(ステップS 1 1 1)。そして、再生成されたハッシュ値h 1'～h 3'と受信した送信分割データDD (1)～DD (3)のハッシュ値h 1～h 3とを比較して一致するか否かを確認する(ステップS 1 1 2)。これにより、各保管サーバ3 a 1～3 a 3から取得したデータの内容が欠落したり、改竄されていないことを確認することができる。

【0096】

次に、ステップS 1 1 2でハッシュ値が一致する送信分割データDD (1)～DD (3)が少なくとも2以上ある場合には、元データ復元部3 6の機能として、分割データD (1)～D (3)およびヘッダ情報A 1～A 3に基づいて秘密分散法Sにより元データB 1およびハッシュ値H 1をそれぞれ復元する(ステップS 1 1 3)。これは、保管サーバ3 a 1～3 a 3から分割データを取得できない場合やハッシュ値の一致が確認できない場合があっても、データが取得でき、ハッシュ値の一致が確認できたものがデータの復元に必要な個数以上(本実施の形態においては、後述するように2つ以上)あった場合には、後続処理(データ復元処理)を行うものである。

【0097】

次いで、分割装置6は、ハッシュ値確認部3 5の機能として、復元した元データB 1からハッシュ値H 2を再生成し、再生成したハッシュ値H 2と復元したハッシュ値H 1とを比較して一致するか否かを確認する(ステップS 1 1 4, S 1 1 5)。これにより、復元された元データBおよびハッシュ値Hの内容が欠落したり、改竄されていないことを確認することができる。また、ヘッダ情報A 1～A 3および分割データ本体D (1)～D (3)の内容を改竄した後、改竄した内容に合わせてハッシュ値h 1～h 3を算出し、書き換えた場合においても(この場合、ステップS 1 1 2においてハッシュ値は一致してしまう)、改竄を検出をすることが可能となる。

【0098】

そして、分割装置6は、復元された元データBをネットワークNを介して端末5に送信し、これにより、端末5は元データBを取得する(ステップS 1 1 6, S 1 1 7)。

【0099】

ここで、本実施の形態における秘密分散法Sについて詳しく説明する。尚、以後の説明における元データとは、データ分割の対象となる元データ、即ち、本実施の形態においては、ハッシュ値生成部3 3で生成されたハッシュ値Hを含む元データBをいう。

【0100】

本実施形態における元データの分割および復元では、元データを所望の処理単位ビット長に基づいて所望の分割数の分割データに分割するが、この場合の処理単位ビット長は任意の値に設定することができ、元データを処理単位ビット長毎に区分けして、この元部分データから分割部分データを分割数より1少ない数ずつ生成するので、元データのビット長が処理単位ビット長の(分割数-1)倍の整数倍に一致しない場合は、元データの末尾の部分に0を埋めるなどして元データのビット長を処理単位ビット長の(分割数-1)倍の整数倍に合わせることで本実施形態を適用することができる。

【0101】

また、上述した乱数も(分割数-1)個の元部分データの各々に対応して処理単位ビット長のビット長を有する(分割数-1)個の乱数部分データとして乱数発生部3 4から生成さ

れる。すなわち、乱数は処理単位ビット長毎に区分けされて、処理単位ビット長のビット長を有する（分割数-1）個の乱数部分データとして生成される。更に、元データは処理単位ビット長に基づいて所望の分割数の分割データに分割されるが、この分割データの各々も（分割数-1）個の元部分データの各々に対応して処理単位ビット長のビット長を有する（分割数-1）個の分割部分データとして生成される。すなわち、分割データの各々は、処理単位ビット長毎に区分けされて、処理単位ビット長のビット長を有する（分割数-1）個の分割部分データとして生成される。

【0102】

なお、以下の説明では、上述した元データ、乱数、分割データ、分割数および処理単位ビット長をそれぞれ B, R, D, n および b で表すとともに、また複数のデータや乱数などのうちの1つを表わす変数として $i (=1 \sim n)$ および $j (=1 \sim n-1)$ を用い、（分割数 $n-1$ ）個の元部分データ、（分割数 $n-1$ ）個の乱数部分データ、および分割数 n 個の分割データ D のそれぞれのうちの1つをそれぞれ $B(j), R(j)$ および $D(i)$ で表記し、更に各分割データ $D(i)$ を構成する複数 $(n-1)$ の分割部分データを $D(i, j)$ で表記するものとする。すなわち、 $B(j)$ は、元データ B の先頭から処理単位ビット長毎に区分けして1番から順に採番した時の j 番目の元部分データを表すものである。

【0103】

この表記を用いると、元データ、乱数データ、分割データとこれらをそれぞれ構成する元部分データ、乱数部分データ、分割部分データは、次のように表記される。

【0104】

元データ $B=(n-1)$ 個の元部分データ $B(j)$
 $=B(1), B(2), \dots, B(n-1)$
 乱数 $R=(n-1)$ 個の乱数部分データ $R(j)$
 $=R(1), R(2), \dots, R(n-1)$
 n 個の分割データ $D(i)=D(1), D(2), \dots, D(n)$
 各分割部分データ $D(i, j)$
 $=D(1, 1), D(1, 2), \dots, D(1, n-1)$
 $D(2, 1), D(2, 2), \dots, D(2, n-1)$
 $\dots \quad \dots \quad \dots$
 $D(n, 1), D(n, 2), \dots, D(n, n-1)$
 $(i=1 \sim n), (j=1 \sim n-1)$

本実施形態は、上述したように処理単位ビット長毎に区分けされる複数の部分データに対して元部分データと乱数部分データの排他的論理和演算（XOR）を行って、詳しくは、元部分データと乱数部分データの排他的論理和演算（XOR）からなる定義式を用いて、元データの分割を行うことを特徴とするものであり、上述したデータ分割処理に多項式や剰余演算を用いる方法（第1および第2の実施の形態における方法）に比較して、コンピュータ処理に適したビット演算である排他的論理和（XOR）演算を用いることにより高速かつ高性能な演算処理能力を必要とせず、大容量のデータに対しても簡単な演算処理を繰り返して分割データを生成することができるとともに、また分割データの保管に必要な記憶容量も分割数に比例した倍数の容量よりも小さくすることができる。更に、任意に定めた一定の長さ毎にデータの先頭から順に演算処理を行うストリーム処理により分割データが生成される。

【0105】

なお、本実施形態で使用する排他的論理和演算（XOR）は、以下の説明では、「 $*$ 」なる演算記号で表すことにするが、この排他的論理和演算のビット毎の演算規則での各演算結果は下記のとおりである。

【0106】

0 * 0 の演算結果は 0
 0 * 1 の演算結果は 1
 1 * 0 の演算結果は 1

1 * 1 の演算結果は 0

また、XOR演算は交換法則、結合法則が成り立つ。すなわち、

$$a*b=b*a$$

$(a*b)*c=a*(b*c)$ が成り立つことが数学的に証明される。

【0107】

また、 $a*a=0$, $a*0=0*a=a$ が成り立つ。

【0108】

ここで a, b, c は同じ長さのビット列を表し、0 はこれらと同じ長さですべて「0」からなるビット列を表す。

【0109】

次に、フローチャートなどの図面も参照して、本実施の形態における秘密分散法 S の作用について説明するが、この説明の前に図 9 乃至 12 のフローチャートに示す記号の定義について説明する。

【0110】

(1) $\prod_{i=1}^n A(i)$ は、 $A(1)*A(2)*\cdots*A(n)$ を意味するものとする。

【0111】

(2) $c(j, i, k)$ を $(n-1) \times (n-1)$ 行列である $U[n-1, n-1] \times (P[n-1, n-1])^{(j-1)}$ の i 行 k 列の値と定義する。

【0112】

このとき $Q(j, i, k)$ を下記のように定義する。

【0113】

$c(j, i, k)=1$ のとき $Q(j, i, k)=R((n-1) \times m+k)$

$c(j, i, k)=0$ のとき $Q(j, i, k)=0$

(3) $U[n, n]$ とは、 $n \times n$ 行列であって、 i 行 j 列の値を $u(i, j)$ で表すと、

$i+j \leq n+1$ のとき $u(i, j)=1$

$i+j > n+1$ のとき $u(i, j)=0$

である行列を意味するものとし、「上三角行列」ということとする。具体的には下記のよ
うな行列である。

【数1】

$$U[3, 3] = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad U[4, 4] = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

【数2】

$$P[3, 3] = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \quad U[4, 4] = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

(5) A, Bを $n \times n$ 行列とすると、 $A \times B$ とは行列AとBの積を意味するものとする。行列の成分同士の計算規則は通常の数学で用いるものと同じである。

【0114】

(6) Aを $n \times n$ 行列とし、iを整数とすると、 A^i とは行列Aのi個の積を意味するものとする。また、 A^0 とは単位行列Eを意味するものとする。

【0115】

(7) 単位行列 $E[n, n]$ とは、 $n \times n$ 行列であって、i行j列の値を $e(i, j)$ で表すと、

i=j のとき $e(i, j)=1$

上記以外のとき $e(i, j)=0$

である行列を意味するものとする。具体的には下記のような行列である。Aを任意の $n \times n$ 行列とすると

$$A \times E = E \times A = A$$

となる性質がある。

【数3】

$$E[3, 3] = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad E[4, 4] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

次に、図9に示すフローチャートおよび図10および図11に示す具体的データなどを参照して、まず元データBの分割処理について説明する。

【0116】

まず、元データBを分割装置6に供給する(図9のステップS201)。なお、本例では、元データBは、16ビットの「10110010 00110111」とする。

【0117】

次に、利用者は端末5から分割数nとして3を分割装置6に指示する(ステップS203)。この分割数nは分割装置6において予め定められた値を用いてもよい。なお、この分割数n=3に従って分割装置6で生成される3個の分割データをD(1)、D(2)、D(3)とする。この分割データD(1)、D(2)、D(3)は、すべて元データのビット長と同じ16ビット長のデータである。

【0118】

それから、元データBを分割するために使用される処理単位ビット長bを8ビットと決定し、また元データと同じビット長である16ビットの乱数Rを乱数発生部34から取得して生成する(ステップS205)。この処理単位ビット長bは、利用者が端末5から分割装置6に対して指定してもよいし、または分割装置6において予め定められた値を用いてもよい。なお、処理単位ビット長bは、任意のビット数でよいが、ここでは元データBを割り切ることができる8ビットとしている。従って、上記16ビットの「10110010 00110111」の元データBは、8ビットの処理単位ビット長で分けられた場合の2個の元分割データB(1)およびB(2)は、それぞれ「10110010」および「00110111」となる。

【0119】

次のステップS 2 0 7では、元データBのビット長が 8×2 の整数倍であるか否かを判定し、整数倍でない場合には、元データBの末尾を0で埋めて、 8×2 の整数倍に合わせる。なお、本例のように処理単位ビット長 b が8ビットおよび分割数 n が3に設定された場合における分割処理は、元データBのビット長として16ビットに限られるものでなく、処理単位ビット長 $b \times (\text{分割数}n-1) = 8 \times 2$ の整数倍の元データBに対して有効なものである。

【0120】

次に、ステップS 2 0 9では、変数 m 、すなわち上述した整数倍を意味する変数 m を0に設定する。本例のように、元データBが処理単位ビット長 $b \times (\text{分割数}n-1) = 8 \times 2 = 16$ ビットである場合には、変数 m は0であるが、2倍の32ビットの場合には、変数 m は1となり、3倍の48ビットの場合には、変数 m は2となる。

【0121】

次に、元データBの $8 \times 2 \times m + 1$ ビット目から 8×2 ビット分のデータが存在するか否かが判定される（ステップS 2 1 1）。これは、このステップS 2 1 1以降に示す分割処理を元データBの変数 m で特定される処理単位ビット長 $b \times (\text{分割数}n-1) = 8 \times 2 = 16$ ビットに対して行った後、元データBとして次の16ビットがあるか否かを判定しているものである。本例のように元データBが16ビットである場合には、16ビットの元データBに対してステップS 2 1 1以降の分割処理を1回行くと、後述するステップS 2 1 9で変数 m が+1されるが、本例の元データBでは変数 m が $m+1$ の場合に相当する17ビット以降のデータは存在しないので、ステップS 2 1 1からステップS 2 2 1に進むことになるが、今の場合は、変数 m は0であるので、元データBの $8 \times 2 \times m + 1$ ビット目は、 $8 \times 2 \times 0 + 1 = 1$ となり、元データBの16ビットの1ビット目から 8×2 ビット分にデータが存在するため、ステップS 2 1 3に進む。

【0122】

ステップS 2 1 3では、変数 j を1から2（=分割数 $n-1$ ）まで変えて、元データBの $8 \times (2 \times m + j - 1) + 1$ ビット目から8ビット分（=処理単位ビット長）のデータを元部分データB($2 \times m + j$)に設定し、これにより元データBを処理単位ビット長で区分けした2（分割数 $n-1$ ）個の元部分データB(1), B(2)を次のように生成する。

【0123】

元データB=B(1), B(2)

第1の元部分データB(1)=「10110010」

第2の元部分データB(2)=「00110111」

次に、変数 j を1から2（=分割数 $n-1$ ）まで変えて、乱数部分データR($2 \times m + j$)に乱数発生部34から発生する8ビットの長さの乱数を設定し、これにより乱数Rを処理単位ビット長で区分けした2（分割数 $n-1$ ）個の乱数部分データR(1), R(2)を次のように生成する（ステップS 2 1 5）。

【0124】

乱数R=R(1), R(2)

第1の乱数部分データR(1)=「10110001」

第2の乱数部分データR(2)=「00110101」

次に、ステップS 2 1 7において、変数 i を1から3（=分割数 n ）まで変えるとともに、更に各変数 i において変数 j を1から2（=分割数 $n-1$ ）まで変えながら、ステップS 2 1 7に示す分割データを生成するための元部分データと乱数部分データの排他的論理和からなる定義式により複数の分割データD(i)の各々を構成する各分割部分データD($i, 2 \times m + j$)を生成する。この結果、次に示すような分割データDが生成される。

【0125】

分割データD

=3個の分割データD(i)=D(1), D(2), D(3)

第1の分割データD(1)

=2個の分割部分データD(1, j)=D(1, 1), D(1, 2)

= 「00110110」, 「10110011」

第2の分割データD(2)

= 2個の分割部分データD(2, j)=D(2, 1), D(2, 2)

= 「00000011」, 「00000010」

第3の分割データD(3)

= 2個の分割部分データD(3, j)=D(3, 1), D(3, 2)

= 「10110001」, 「00110101」

なお、各分割部分データ(i, j)を生成するためのステップS 2 1 7に示す定義式は、本例のように分割数n=3の場合には、具体的には図1 1に示す表に記載されているものとなる。図1 1に示す表から、分割部分データD(1, 1)を生成するための定義式は $B(1)*R(1)*R(2)$ であり、D(1, 2)の定義式は $B(2)*R(1)*R(2)$ であり、D(2, 1)の定義式は $B(1)*R(1)$ であり、D(2, 2)の定義式は $B(2)*R(2)$ であり、D(3, 1)の定義式は $R(1)$ であり、D(3, 2)の定義式は $R(2)$ である。また、図1 1に示す表には $m>0$ の場合の任意の整数についての一般的な定義式も記載されている。

【0126】

このように整数倍を意味する変数 $m=0$ の場合について分割データDを生成した後、次に変数 m を1増やし（ステップS 2 1 9）、ステップS 2 1 1に戻り、変数 $m+1$ に該当する元データBの1 7ビット以降について同様の分割処理を行おうとするが、本例の元データBは1 6ビットであり、1 7ビット以降のデータは存在しないので、ステップS 2 1 1からステップS 2 2 1に進み、上述したように生成した分割データD(1), D(2), D(3)を分割装置6のメモリ1 0に一時保存して、分割処理を終了する。なお、このように保管された分割データD(1), D(2), D(3)はそれぞれ単独では元データが推測できない。

【0127】

ここで、上述した図9のフローチャートのステップS 2 1 7における定義式による分割データの生成処理、具体的には分割数n=3の場合の分割データの生成処理について詳しく説明する。

【0128】

まず、整数倍を意味する変数 $m=0$ の場合には、ステップS 2 1 7に示す定義式から各分割データ $D(i)=D(1)\sim D(3)$ の各々を構成する各分割部分データ $D(i, 2\times m+j)=D(i, j)$ ($i=1\sim 3$, $j=1\sim 2$)は、次のようになる。

【0129】

$D(1, 1)=B(1)*Q(1, 1, 1)*Q(1, 1, 2)$

$D(1, 2)=B(2)*Q(2, 1, 1)*Q(2, 1, 2)$

$D(2, 1)=B(1)*Q(1, 2, 1)*Q(1, 2, 2)$

$D(2, 2)=B(2)*Q(2, 2, 1)*Q(2, 2, 2)$

$D(3, 1)=R(1)$

$D(3, 2)=R(2)$

上記の6つの式のうち上から4つの式に含まれる $Q(j, i, k)$ を具体的に求める。

【0130】

これは $c(j, i, k)$ を 2×2 行列である $U[2, 2]\times (P[2, 2])^{j-1}$ の i 行 k 列の値としたとき下記のように定義される。

【0131】

$c(j, i, k)=1$ のとき $Q(j, i, k)=R(k)$

$c(j, i, k)=0$ のとき $Q(j, i, k)=0$

ここで、

$j=1$ のときは

【数4】

$$\begin{aligned}
 U[2, 2] \times (P[2, 2])^{\wedge (j-1)} &= U[2, 2] \times (P[2, 2])^{\wedge 0} \\
 &= U[2, 2] \times E[2, 2] \\
 &= U[2, 2] \\
 &= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}
 \end{aligned}$$

【数5】

$$\begin{aligned}
 U[2, 2] \times (P[2, 2])^{\wedge (j-1)} &= U[2, 2] \times (P[2, 2])^{\wedge 1} \\
 &= U[2, 2] \times P[2, 2] \\
 &= \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \\
 &= \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}
 \end{aligned}$$

これを用いると、各分割部分データD(i, j)は次のような定義式により生成される。

【0132】

$$\begin{aligned}
 D(1, 1) &= B(1) * Q(1, 1, 1) * Q(1, 1, 2) = B(1) * R(1) * R(2) \\
 D(1, 2) &= B(2) * Q(2, 1, 1) * Q(2, 1, 2) = B(2) * R(1) * R(2) \\
 D(2, 1) &= B(1) * Q(1, 2, 1) * Q(1, 2, 2) = B(1) * R(1) * 0 = B(1) * R(1) \\
 D(2, 2) &= B(2) * Q(2, 2, 1) * Q(2, 2, 2) = B(2) * 0 * R(2) = B(2) * R(2)
 \end{aligned}$$

上述した各分割部分データD(i, j)を生成するための定義式は、図10にも図示されている。

【0133】

図10は、上述したように16ビットの元データBを8ビットの処理単位ビット長に基づいて分割数n=3で3分割する場合の各データと定義式および各分割部分データから元データを復元する場合の計算式などを示す表である。

【0134】

ここで、上述した定義式により分割データD(1), D(2), D(3)および各分割部分データD(1, 1), D(1, 2), D(2, 1), D(2, 2), D(3, 1), D(3, 2)を生成する過程と定義式の一般形について説明する。

【0135】

まず、第1の分割データD(1)に対しては、第1の分割部分データD(1, 1)は、上述した定義式B(1)*R(1)*R(2)で定義され、第2の分割部分データD(1, 2)は定義式B(2)*R(1)*R(2)で定義される。なお、この定義式の一般形は、D(1, j)に対してはB(j)*R(j)*R(j+1)であり、D(1, j+1)に対してB(j+1)*R(j)*R(j+1)である(jは奇数とする)。定義式に従って計算すると、D(1, 1)は00110110、D(1, 2)は10110011となるので、D(1)は00110110 10110011である。なお、定義式の一般形は、図11にまとめて示されている。

【0136】

また、第2の分割データD(2)に対しては、D(2, 1)はB(1)*R(1)で定義され、D(2, 2)はB(2)

) $\cdot R(2)$ で定義される。この定義式の一般形は、 $D(2, j)$ に対しては $B(j) \cdot R(j)$ であり、 $D(2, j+1)$ に対しては $B(j+1) \cdot R(j+1)$ である (j は奇数とする)。定義式に従って計算すると、 $D(2, 1)$ は00000011, $D(2, 2)$ は00000010となるので、 $D(2)$ は00000011 00000010である。

【0137】

更に第3の分割データ $D(3)$ に対しては、 $D(3, 1)$ は $R(1)$ で定義され、 $D(3, 2)$ は $R(2)$ で定義される。この定義式の一般形は、 $D(3, j)$ に対しては $R(j)$ であり、 $D(3, j+1)$ に対しては $R(j+1)$ である (j は奇数とする)。定義式に従って計算すると、 $D(3, 1)$ は10110001, $D(3, 2)$ は00110101となるので、 $D(3)$ は10110001 00110101である。

【0138】

上記説明は、 $B, R, D(1), D(2), D(3)$ の長さを16ビットとしたが、データの先頭から上記分割処理を繰り返すことにより、どのような長さの元データ B からでも分割データ $D(1), D(2), D(3)$ を生成することができる。また、処理単位ビット長 b は任意にとることができ、元データ B の先頭から順に $b \times 2$ の長さ毎に上記分割処理を繰り返すことにより任意の長さの元データ、具体的には処理単位ビット長 $b \times 2$ の整数倍の長さの元データに対して適用することができる。なお、元データ B の長さが処理単位ビット長 $b \times 2$ の整数倍でない場合は、例えば、データ末尾の部分を0で埋めるなどして元データ B の長さを処理単位ビット長 $b \times 2$ の整数倍に合わせるることにより上述した本実施形態の分割処理を適用することができる。

【0139】

次に、図10の右側に示す表を参照して、分割データから元データを復元する処理について説明する。

【0140】

まず、分割装置6に元データ B の復元を要求する。分割装置6は、この元データ B の復元要求を受け取ると、この元データ B に対応する分割データ $D(1), D(2), D(3)$ が保管サーバ3 a 1, 3 a 2, 3 a 3に保管されていることを記憶しているので、ネットワーク N を介して保管サーバ3 a 1, 3 a 2, 3 a 3から分割データ $D(1), D(2), D(3)$ を取得し、この取得した分割データ $D(1), D(2), D(3)$ から次に示すように元データ B を復元する。

【0141】

まず、分割部分データ $D(2, 1), D(3, 1)$ から第1の元部分データ $B(1)$ を次のように生成することができる。

【0142】

$$\begin{aligned} D(2, 1) \cdot D(3, 1) &= (B(1) \cdot R(1)) \cdot R(1) \\ &= B(1) \cdot (R(1) \cdot R(1)) \\ &= B(1) \cdot 0 \\ &= B(1) \end{aligned}$$

具体的に計算すると、 $D(2, 1)$ は00000011, $D(3, 1)$ は10110001なので、 $B(1)$ は10110010となる。

【0143】

また、別の分割部分データから次のように第2の元部分データ $B(2)$ を生成することができる。

【0144】

$$\begin{aligned} D(2, 2) \cdot D(3, 2) &= (B(2) \cdot R(2)) \cdot R(2) \\ &= B(2) \cdot (R(2) \cdot R(2)) \\ &= B(2) \cdot 0 \\ &= B(2) \end{aligned}$$

具体的に計算すると、 $D(2, 2)$ は00000010, $D(3, 2)$ は00110101なので、 $B(2)$ は00110111となる。

【0145】

一般に、 j を奇数として、

$$D(2, j) \cdot D(3, j) = (B(j) \cdot R(j)) \cdot R(j)$$

$$\begin{aligned}
 &=B(j)*(R(j)*R(j)) \\
 &=B(j)*0 \\
 &=B(j)
 \end{aligned}$$

であるから、 $D(2,j)*D(3,j)$ を計算すれば、 $B(j)$ が求まる。

【0146】

また、一般に、 j を奇数として、

$$\begin{aligned}
 D(2,j+1)*D(3,j+1) &= (B(j+1)*R(j+1))*R(j+1) \\
 &= B(j+1)*(R(j+1)*R(j+1)) \\
 &= B(j+1)*0 \\
 &= B(j+1)
 \end{aligned}$$

であるから、 $D(2,j+1)*D(3,j+1)$ を計算すれば、 $B(j+1)$ が求まる。

【0147】

次に、 $D(1), D(3)$ を取得して B を復元する場合には、次のようになる。

【0148】

$$\begin{aligned}
 D(1,1)*D(3,1)*D(3,2) &= (B(1)*R(1)*R(2))*R(1)*R(2) = B(1)*(R(1)*R(1))*(R(2)*R(2)) \\
 &= B(1)*0*0 \\
 &= B(1)
 \end{aligned}$$

であるから、 $D(1,1)*D(3,1)*D(3,2)$ を計算すれば、 $B(1)$ が求まる。具体的に計算すると、 $D(1,1)$ は00110110、 $D(3,1)$ は10110001、 $D(3,2)$ は00110101なので、 $B(1)$ は10110010となる。

【0149】

また同様に、

$$\begin{aligned}
 D(1,2)*D(3,1)*D(3,2) &= (B(2)*R(1)*R(2))*R(1)*R(2) \\
 &= B(2)*(R(1)*R(1))*(R(2)*R(2)) \\
 &= B(2)*0*0 \\
 &= B(2)
 \end{aligned}$$

であるから、 $D(1,2)*D(3,1)*D(3,2)$ を計算すれば、 $B(2)$ が求まる。具体的に計算すると、 $D(1,2)$ は10110011、 $D(3,1)$ は10110001、 $D(3,2)$ は00110101なので、 $B(2)$ は00110111となる。

【0150】

一般に、 j を奇数として、

$$\begin{aligned}
 D(1,j)*D(3,j)*D(3,j+1) &= (B(j)*R(j)*R(j+1))*R(j)*R(j+1) \\
 &= B(j)*(R(j)*R(j))*(R(j+1)*R(j+1)) \\
 &= B(j)*0*0 \\
 &= B(j)
 \end{aligned}$$

であるから、 $D(1,j)*D(3,j)*D(3,j+1)$ を計算すれば、 $B(j)$ が求まる。

【0151】

また、一般に、 j を奇数として、

$$\begin{aligned}
 D(1,j+1)*D(3,j)*D(3,j+1) &= (B(j+1)*R(j)*R(j+1))*R(j)*R(j+1) \\
 &= B(j+1)*(R(j)*R(j))*(R(j+1)*R(j+1)) \\
 &= B(j+1)*0*0 \\
 &= B(j+1)
 \end{aligned}$$

であるから、 $D(1,j+1)*D(3,j)*D(3,j+1)$ を計算すれば、 $B(j+1)$ が求まる。

【0152】

次に、 $D(1), D(2)$ を取得して B を復元する場合には、次のようになる。

【0153】

$$\begin{aligned}
 D(1,1)*D(2,1) &= (B(1)*R(1)*R(2))*(B(1)*R(1)) \\
 &= (B(1)*B(1))*(R(1)*R(1))*R(2) \\
 &= 0*0*R(2) \\
 &= R(2)
 \end{aligned}$$

であるから、 $D(1,1)*D(2,1)$ を計算すれば、 $R(2)$ が求まる。具体的に計算すると、 $D(1,1)$ は00110110、 $D(2,1)$ は00000011なので、 $R(2)$ は00110101となる。

【0154】

また同様に、

$$\begin{aligned} D(1,2)*D(2,2) &= (B(2)*R(1)*R(2))*(B(2)*R(2)) \\ &= (B(2)*B(2))*R(1)*(R(2)*R(2)) \\ &= 0*R(1)*0 \\ &= R(1) \end{aligned}$$

であるから、 $D(1,2)*D(2,2)$ を計算すれば、 $R(1)$ が求まる。具体的に計算すると、 $D(1,2)$ は10110011、 $D(2,2)$ は00000010なので、 $R(1)$ は10110001となる。

【0155】

この $R(1)$ 、 $R(2)$ を使用して $B(1)$ 、 $B(2)$ を求める。

【0156】

$$\begin{aligned} D(2,1)*R(1) &= (B(1)*R(1))*R(1) \\ &= B(1)*(R(1)*R(1)) \\ &= B(1)*0 \\ &= B(1) \end{aligned}$$

であるから、 $D(2,1)*R(1)$ を計算すれば、 $B(1)$ が求まる。具体的に計算すると、 $D(2,1)$ は00000011、 $R(1)$ は10110001なので、 $B(1)$ は10110010となる。

【0157】

また同様に、

$$\begin{aligned} D(2,2)*R(2) &= (B(2)*R(2))*R(2) \\ &= B(2)*(R(2)*R(2)) \\ &= B(2)*0 \\ &= B(2) \end{aligned}$$

であるから $D(2,2)*R(2)$ を計算すれば $B(2)$ が求まる。具体的に計算すると $D(2,2)$ は00000001 0、 $R(2)$ は00110101なので、 $B(2)$ は00110111となる。

【0158】

一般に、 j を奇数として、

$$\begin{aligned} D(1,j)*D(2,j) &= (B(j)*R(j)*R(j+1))*(B(j)*R(j)) \\ &= (B(j)*B(j))*(R(j)*R(j))*R(j+1) \\ &= 0*0*R(j+1) \\ &= R(j+1) \end{aligned}$$

であるから $D(1,j)*D(2,j)$ を計算すれば $R(j+1)$ が求まる。

【0159】

また同様に、

$$\begin{aligned} D(1,j+1)*D(2,j+1) &= (B(j+1)*R(j)*R(j+1))*(B(j+1)*R(j+1)) \\ &= (B(j+1)*B(j+1))*R(j)*(R(j+1)*R(j+1)) \\ &= 0*R(j)*0 \\ &= R(j) \end{aligned}$$

であるから $D(1,j+1)*D(2,j+1)$ を計算すれば $R(j)$ が求まる。

【0160】

この $R(j)$ 、 $R(j+1)$ を使用して $B(j)$ 、 $B(j+1)$ を求める。

【0161】

$$\begin{aligned} D(2,j)*R(j) &= (B(j)*R(j))*R(j) \\ &= B(j)*(R(j)*R(j)) \\ &= B(j)*0 \\ &= B(j) \end{aligned}$$

であるから $D(2,j)*R(j)$ を計算すれば $B(j)$ が求まる。

【0162】

また同様に、

$$\begin{aligned} D(2, j+1) * R(j+1) &= (B(j+1) * R(j+1)) * R(j+1) \\ &= B(j+1) * (R(j+1) * R(j+1)) \\ &= B(j+1) * 0 \\ &= B(j+1) \end{aligned}$$

であるから $D(2, j+1) * R(j+1)$ を計算すれば $B(j+1)$ が求まる。

【0163】

上述したように、元データの先頭から処理単位ビット長 b に基づいて分割処理を繰り返して行くと、分割データを生成した場合には、3つの分割データ $D(1), D(2), D(3)$ のすべてを用いなくても、3つの分割データのうち、2つの分割データを用いて上述したように元データを復元することができる。

【0164】

本発明の他の方法として、乱数 R のビット長を元データ B のビット長よりも短いものを使用して、元データの分割処理を行うことができる。

【0165】

すなわち、上述した乱数 R は $B, D(1), D(2), D(3)$ と同じビット長のデータとしたが、乱数 R を元データ B のビット長より短いものとし、分割データ $D(1), D(2), D(3)$ の生成にこの短いビット長の乱数 R を繰り返し用いるものである。

【0166】

なお、分割データ $D(3)$ は乱数 R のみから生成されるので、分割データ $D(3)$ は乱数 R を繰り返して保管しておく必要はない。例えば、元データ B のビット長を 1600 ビット (200 バイト) としたとき、乱数 R は任意にとった 160 ビット (20 バイト) のデータの繰り返しとする。つまり、 $R(1) \sim R(20)$ はランダムに生成し、 $R(21) \sim R(200)$ は $R(21) = R(1), R(22) = R(2), \dots, R(40) = R(20), R(41) = R(1), R(42) = R(2), \dots, R(60) = R(20), R(61) = R(1), R(62) = R(2), \dots, R(80) = R(20), \dots, R(181) = R(1), R(182) = R(2), \dots, R(200) = R(20)$ とする。

【0167】

先の説明では、分割部分データ $D(3, j)$ を乱数部分データ $R(j)$ と定義して $D(3)$ を生成しているが、 $D(3, 20)$ まで保管すれば十分である。つまり、 $D(3)$ の長さは $D(1), D(2)$ の 10 分の 1 となる。従って、保管すべきデータの総量は先の実施形態では元データ B の 3 倍であるが、この実施形態では 2.1 倍とすることができる。

【0168】

乱数 R における繰り返し部分のデータの長さは、短すぎると、 $D(1)$ または $D(2)$ のみから R が解読されてしまうことも考えられるため、適切な長さを選択することが望ましい。

【0169】

次に、図 12 に示すフローチャートを参照して、分割数が n で、処理単位ビット長が b である場合の一般的な分割処理について説明する。

【0170】

まず、元データ B を分割装置 6 に供給する (ステップ S401)。また、利用者は端末 5 から分割数 n ($n \geq 3$ である任意の整数) を分割装置 6 に指示する (ステップ S403)。この分割数 n は分割装置 6 において予め定められた値を用いてもよい。処理単位ビット長 b を決定する (ステップ S405)。なお、 b は 0 より大きい任意の整数である。次に、元データ B のビット長が $b \times (n-1)$ の整数倍であるかを判定し、整数倍でない場合には、元データ B の末尾を 0 で埋める (ステップ S407)。また、整数倍を意味する変数 m を 0 に設定する (ステップ S409)。

【0171】

次に、元データ B の $b \times (n-1) \times m + 1$ ビット目から $b \times (n-1)$ ビット分のデータが存在するかが判定される (ステップ S411)。この判定の結果、データが存在しない場合は、ステップ S421 に進むことになるが、今の場合は、ステップ S409 で変数 m は 0 に設定された場合であるので、データが存在するため、ステップ S413 に進む。

【0172】

ステップS 4 1 3では、変数jを1からn-1まで変えて、元データBの $b \times (n-1) \times m + j - 1$ + 1ビット目からbビット分のデータを元部分データB((n-1)×m+j)に設定する処理を繰り返す、これにより元データBを処理単位ビット長bで区分けした(n-1)個の元部分データB(1), B(2), …B(n-1)が生成される。

【0173】

次に、変数jを1からn-1まで変えて、乱数部分データR((n-1)×m+j)に乱数発生部34から発生する処理単位ビット長bの乱数を設定し、これにより乱数Rを処理単位ビット長bで区分けしたn-1個の乱数部分データR(1), R(2), …R(n-1)が生成される（ステップS 4 1 5）。

【0174】

次に、ステップS 4 1 7において、変数iを1からnまで変えると同時に、更に各変数iにおいて変数jを1からn-1まで変えながら、ステップS 4 1 7に示す分割データを生成するための定義式により複数の分割データD(i)の各々を構成する各分割部分データD(i, (n-1)×m+j)を生成する。この結果、次に示すような分割データDが生成される。

【0175】

分割データD

=n個の分割データD(i)=D(1), D(2), …D(n)

第1の分割データD(1)

=n-1個の分割部分データD(1, j)=D(1, 1), D(1, 2), …D(1, n-1)

第2の分割データD(2)

=n-1個の分割部分データD(2, j)=D(2, 1), D(2, 2), …D(2, n-1)

… … …

… … …

第nの分割データD(n)

=n-1個の分割部分データD(n, j)=D(n, 1), D(n, 2), …D(n, n-1)

このように変数m=0の場合について分割データDを生成した後、次に変数mを1増やし（ステップS 4 1 9）、ステップS 4 1 1に戻り、変数m=1に該当する元データBの $b \times (n-1)$ ビット以降について同様の分割処理を行う。最後にステップS 4 1 1の判定の結果、元データBにデータがなくなった場合、ステップS 4 1 1からステップS 4 2 1に進み、上述したように生成した分割データD(1), …, D(n)を分割装置6のメモリ10に一時保存して、分割処理を終了する。

【0176】

以上述べたように、本実施形態によれば、元データBを、元データBのハッシュ値Hを含めて秘密分散法Sにより複数の分割データD(1)～D(3)に分割し、また、分割データD(1)～D(3)、ヘッダ情報A1～A3、並びに分割データD(1)～D(3)およびヘッダ情報A1～A3から算出されたハッシュ値h1～h3から構成される送信分割データDD(1)～DD(3)を生成し、該送信分割データDD(1)～DD(3)を保管サーバ3a1～3a3にそれぞれ保管しているため、その送信分割データDD(1)～DD(3)の内の少なくとも一部が改竄された場合でも、改竄データを含む送信分割データDD(1)～DD(3)から、その送信分割データDD(1)～DD(3)に対する改竄の有無を容易に確認することができる。

【0177】

この結果、例えば元データBを長期間に亘って保管する場合であっても、電子署名のように、電子証明書の再発行処理および電子署名付きデータ作成処理を繰り返すことなく、簡易に元データBを保管することができる。

【0178】

また、本実施形態によれば、元データBを送信分割データDD(1)～DD(3)に分割して保管サーバ3a1～3a3に保管しているため、例えば、送信分割データDD(1)～DD(3)の内の1つの分割データが改竄された場合でも、残りの分割データを用いて元データBを復元することができる。

【0179】

この結果、元データBの原本性をより確実に確保することができる

また、第1および第2の実施の形態における秘密分散法は、公開鍵暗号方式の秘密鍵の分割管理などで利用されており、比較的データ容量の小さいデータに対して実用的な方法であるが、本実施の形態における秘密分散法Sは、多項式演算・剰余演算などを含む多倍長整数の演算処理を必要としないので、大容量データを多数処理する場合においても簡単かつ迅速にデータの分割および復元を行うことができるという効果を得ることができる。

【0180】

なお、第1および第2の実施の形態では、元データBを分割した分割データD(1)～D(4)を、クライアント2に対してネットワークNを介して接続された保管サーバ3a1～3a4に保管し、第3の実施の形態では、元データBを分割した送信分割データDD(1)～DD(3)を、分割装置6に対してネットワークNを介して接続された保管サーバ3a1～3a3に保管するように構成したが、本発明はこの構成に限定されるものではない。

【0181】

例えば、データ原本性確保システム1の変形例を図13に示す。

【0182】

図13に示すように、変形例に係わるデータ原本性確保システム1Aは、クライアント側単独でのクライアントシステム25として構成されており、このクライアントシステム25は、クライアント2と、このクライアント2に対してLAN等を介して接続された互いにハードウェア的に独立した複数の保管用記憶装置20a1～20a4とを備えている。

【0183】

図13に示すようにデータ原本性確保システム1Aを構成しても、元データBを分割データD(1)～D(4)として複数の保管用記憶装置20a1～20a4に保管することができ、元データBの原本性を簡易かつ確実に確保しながら保管することができる。

【0184】

また、同様に、データ原本性確保システム4の変形例として、端末5、分割装置6、および保管サーバ3a1～3a3をクライアント側単独でのクライアントシステムとして構成してもよいものである。

【0185】

さらに、上記実施の形態で説明した各システム構成は、あくまでも好適な一具体例を示したものであり、機能的に同一であるのならば、他のシステム構成、例えば、データ原本性確保システム1のシステム構成をデータ原本性確保システム4に示すようなシステム構成としてもよい。これは、データ原本性確保システム1のクライアント2の機能を端末5と分割装置6に機能分散したシステム構成とするものである。同様にして、例えば、データ原本性確保システム4のシステム構成をデータ原本性確保システム1に示すようなシステム構成としてもよい。これは、データ原本性確保システム4の端末5と分割装置6の機能を集約してクライアント2の機能とするものである。

【0186】

また、上記実施の形態、ならびにその変形例によれば、秘密分散法としてShamirの秘密分散法{(k, n) 閾値法; 但し、分割数を表すnを4とし、復元できる数を表すkを3とする}、および秘密分散法Sを用いたが、本発明はこの構成に限定されるものではなく、上記実施の形態で述べた秘密分散法以外の他の秘密分散法を用いることも可能である。

【0187】

さらに、上記実施の形態、ならびにその変形例によれば、元データのユニークに識別できる識別データとしてハッシュ値を用いたが、ハッシュ値以外の識別データ、例えばパリティ、チェックサム(元データをブロックに分割し、分割されたブロックを数値とみなして全てのブロックを合計した値)等であってもよい。

【図面の簡単な説明】

【0188】

【図1】本発明の第1の実施の形態に係わるデータ原本性確保システムの概略構成を示すブロック図。

【図2】本発明の第1の実施の形態に係わるデータ原本性確保システムにおけるデータ原本性確保プログラムに基づくデータ原本性確保処理の一例を示す概略フローチャート。

【図3】図2に示すデータ原本性確保処理におけるデータシーケンスを概略的に示す図。

【図4】本発明の第2の実施の形態に係わるデータ原本性確保システムの概略構成を示すブロック図。

【図5】本発明の第3の実施の形態に係わるデータ原本性確保システムの概略構成を示すブロック図。

【図6】本発明の第3の実施の形態に係わるデータ原本性確保システムにおけるデータ原本性確保プログラムに基づくデータ原本性確保処理の一例を示す概略フローチャート。

【図7】図6に示すデータ原本性確保処理におけるデータシーケンスを概略的に示す図。

【図8】本発明の第3の実施の形態における分割データの構成を示す図。

【図9】本発明の第3の実施の形態における分割数 $n=3$ の場合の分割処理を示すフローチャート。

【図10】本発明の第3の実施の形態において、16ビットの元データを8ビットの処理単位ビット長に基づいて分割数 $n=3$ で3分割する場合の各データと定義式および各分割部分データから元データを復元する場合の計算式などを示す表。

【図11】本発明の第3の実施の形態において、分割数 $n=3$ の場合の分割データ、分割部分データ、各分割部分データを生成する定義式を示す表。

【図12】本発明の第3の実施の形態における分割数が n で単位ビット長が b である場合の一般的な分割処理を示すフローチャート。

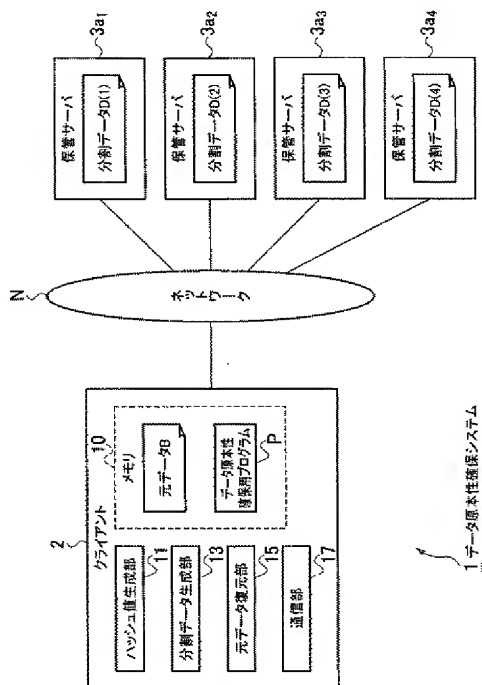
【図13】本発明の変形例に係わるデータ原本性確保システムの概略構成を示すブロック図。

【符号の説明】

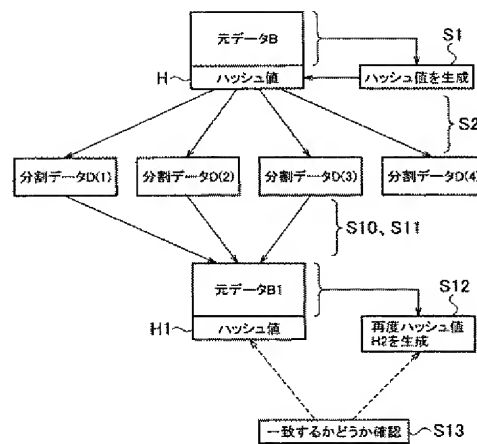
【0189】

- 1、1A、4…データ原本性確保システム
- 2…クライアント
- 3a1～3a4…保管サーバ
- 5…端末
- 6…分割装置
- 10…メモリ
- 11…ハッシュ値生成部
- 13…分割データ生成部
- 15…元データ復元部
- 17…通信部
- 20a1～20a4…保管用記憶装置
- 25…クライアントシステム
- 31、37…データ送受信部
- 32…ハッシュ値生成部
- 33…分割データ生成部
- 34…乱数発生部
- 35…ハッシュ値確認部
- 36…元データ復元部
- 37…データ送受信部

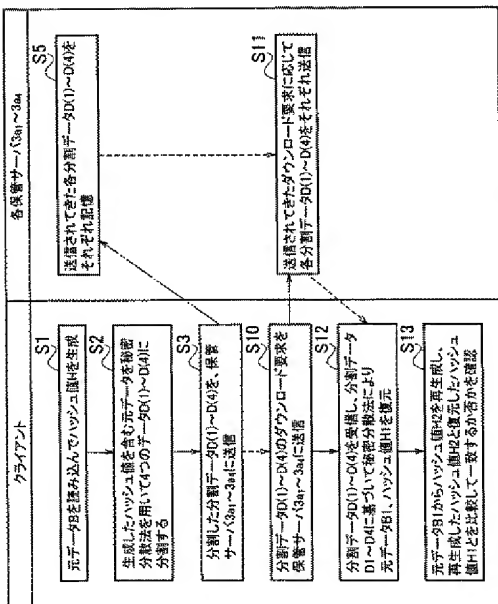
【図1】



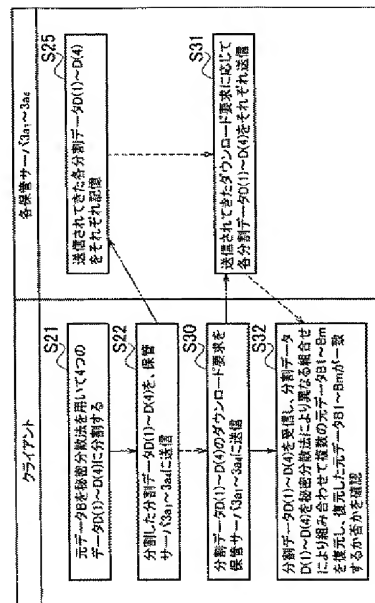
【図2】



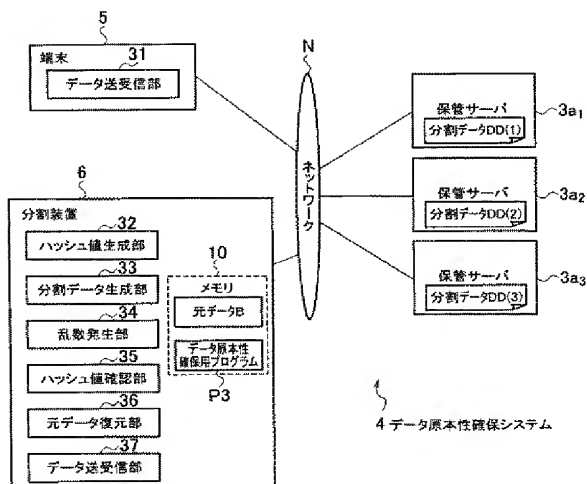
【図3】



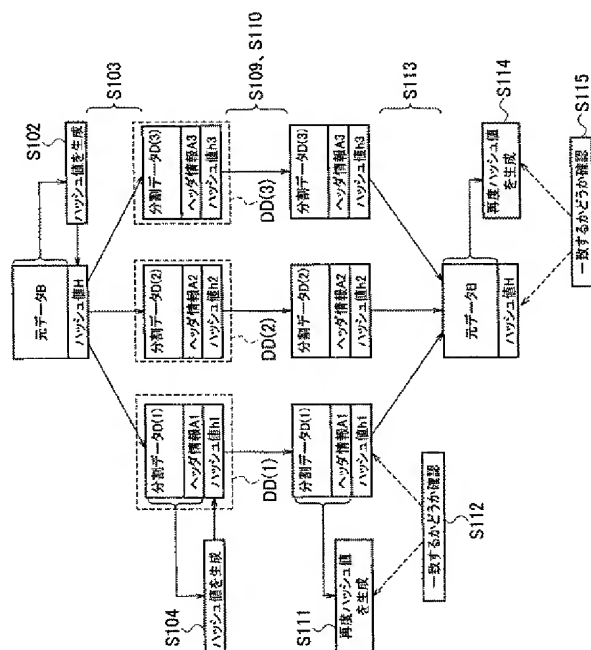
【図4】



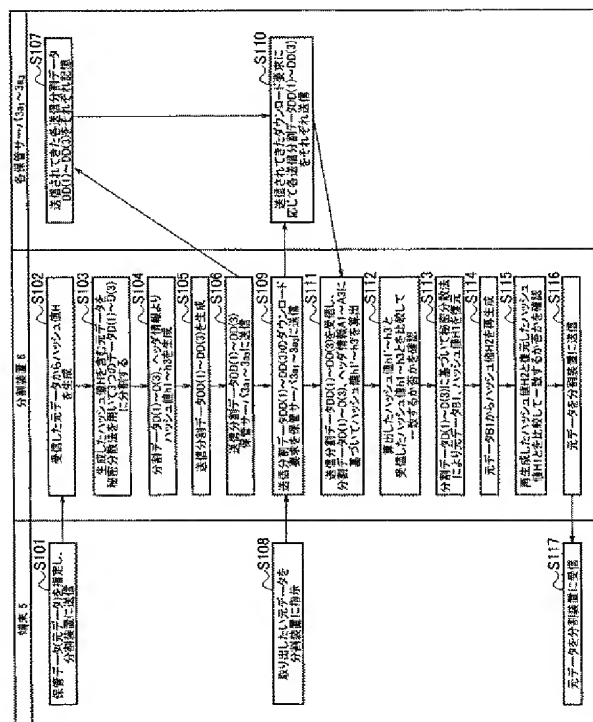
【図5】



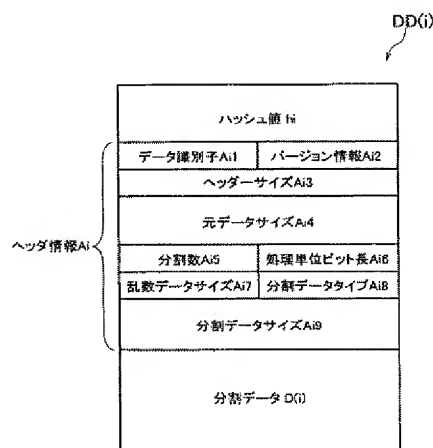
【図6】



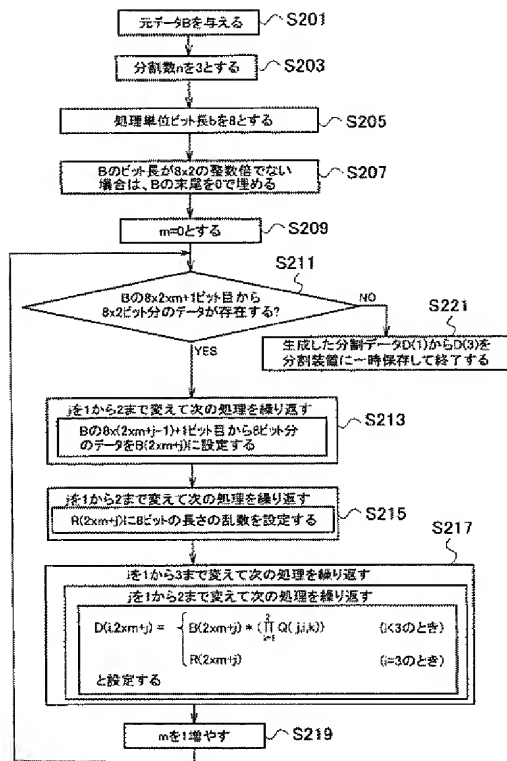
【図7】



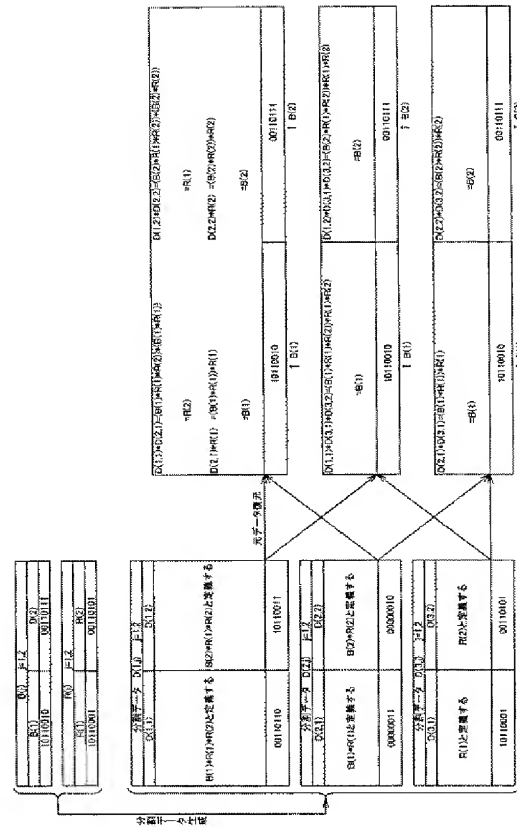
【図8】



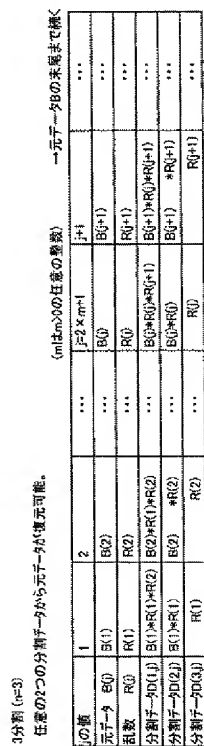
【図9】



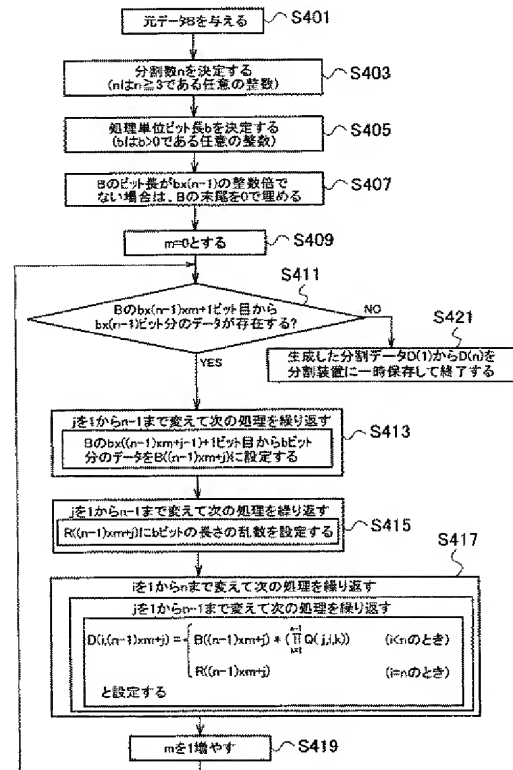
【図10】



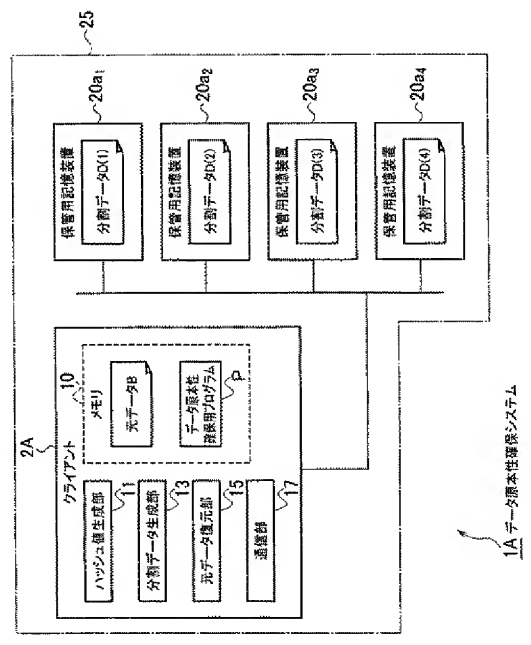
【図11】



【図12】



【図13】



- (72)発明者 萩原 利彦
東京都千代田区内幸町一丁目1番6号 エヌ・ティ・ティ・コミュニケーションズ株式会社内
- (72)発明者 渡部 勝年
東京都千代田区内幸町一丁目1番6号 エヌ・ティ・ティ・コミュニケーションズ株式会社内
- (72)発明者 野村 進
東京都千代田区内幸町一丁目1番6号 エヌ・ティ・ティ・コミュニケーションズ株式会社内
- Fターム(参考) 5J104 AA12